STEPPING & SERVO MOTOR CONTROLLER'S OPTION

MPL-28-01V1.00/USBW32

取扱説明書
(設計者用)

(デバイスドライバ AL MCC05 ユニット編)

CB-23/USB
C-770AL
CDB-5420-AL770
CAD-5410-AL770
C-771
CD-773/ADB5331A

USER'S MANUAL

本製品を使用する前に、この取扱説明書を良く読んで
十分に理解してください。
この取扱説明書は、いつでも取り出して読めるように
保管してください。

MN0143

目次

1. 概要 …… 4

2. 取扱説明書の構成 …… 4

3. ご使用になる前に …… 5
3-1. DLL の使用上の注意 …… 5
3-2. この取扱説明書の見方 …… 6
3-3. マルチプロセス対応 …… 7
3-4. パソコンと I/F ユニット間の通信エラー …… 7
3-5. AL 通信上の通信エラー …… 7

4. AL I/F ユニット対応 DLL …… 8
4-1. 環境設定 …… 8
4-2. 構造体と関数 …… 8
RESULT 構造体 …… 9
環境設定ツール情報構造体 …… 11
スレーブ情報構造体 …… 12
環境設定ツール関数 …… 13
環境設定直接関数 …… 13
環境設定ツール情報読み出し関数 …… 14
環境設定問い合わせ関数 …… 14
AL 通信エラー累計回数読み出し関数 …… 15
AL 通信エラー累計回数クリア関数 …… 15
USB 通信エラー累計回数読み出し関数 …… 16
USB 通信エラー累計回数クリア関数 …… 16
スレーブ情報読み出し関数 …… 17

5. AL MCC05 ユニット対応 DLL …… 18
5-1. デバイスの制御 …… 18
5-2. 汎用 I/O PORT の制御 …… 18
5-3. モータ制御用 I/O の制御 …… 19
5-4. 構造体と関数 …… 20
RESULT 構造体 …… 21
データ構造体 …… 23
デバイスオープン関数 …… 24
デバイスクローズ関数 …… 25
DRIVE COMMAND 一括書き込み関数 …… 25
DRIVE DATA PORT 一括書き込み関数 …… 26
DRIVE COMMAND PORT 書き込み関数 …… 26
DRIVE DATA1 PORT 書き込み関数 …… 27
DRIVE DATA2 PORT 書き込み関数 …… 27
DRIVE DATA3 PORT 書き込み関数 …… 28
STATUS1 PORT 読み出し関数 …… 28
STATUS2 PORT 読み出し関数 …… 29
STATUS3 PORT 読み出し関数 …… 29
STATUS4 PORT 読み出し関数 …… 30
STATUS5 PORT 読み出し関数 …… 30
DRIVE DATA PORT 一括読み出し関数 …… 31
DRIVE DATA1 PORT 読み出し関数 …… 31
DRIVE DATA2 PORT 読み出し関数 …… 32
DRIVE DATA3 PORT 読み出し関数 …… 32
READY WAIT 関数 …… 33
READY WAIT 状態読み出し関数 …… 33
READY WAIT 中止関数 …… 34
COUNTER COMMAND 一括書き込み関数 …… 34
COUNTER COMMAND PORT 書き込み関数 …… 35
COUNTER DATA1 PORT 書き込み関数 …… 35

COUNTER DATA2 PORT 書き込み関数 …… 36
COUNTER DATA3 PORT 書き込み関数 …… 36
汎用 I/O PORT オープン関数 …… 37
汎用 I/O PORT クローズ関数 …… 37
汎用 I/O PORT 一括読み出し関数 …… 38
汎用 I/O PORT 指定信号読み出し関数 …… 39
汎用 I/O PORT 一括書き込み関数 …… 40
汎用 I/O PORT 指定信号 LOW レベル書き込み関数 …… 41
汎用 I/O PORT 指定信号 HIGH レベル書き込み関数 …… 42
制御信号オープン関数 …… 43
制御信号クローズ関数 …… 44
制御信号読み出し関数 …… 45
制御信号指定ビット読み出し関数 …… 46
制御信号書き込み関数 …… 47
制御信号指定ビット ON 書き込み関数 …… 48
制御信号指定ビット OFF 書き込み関数 …… 49
データセット関数 …… 50
データゲット関数 …… 51

6. ソフト開発に必要なファイル …… 52

7. サンプルプログラム …… 53
7-1. 概要 …… 53
7-2. ご使用になる前に …… 53
7-3. 仕様 …… 54

8. トラブルシューティング …… 55

本版で改訂された主な箇所

1. 概要

MPL-28-01V1.00/USBW32 は、DOS/V パソコンの Windows 上で CB-23/USB（AL シリーズ対応 USB I/F ユニット）を使用し、AL シリーズ(弊社オリジナルステッピング＆サーボモータコントローラシステム）上のスレーブユニットを動作させるための DLL ベースの関数ライブラリです。
各関数は、次に示すスレーブユニット上の PORT のアクセス(読み出し/書き込み)を行う為のものです。スレーブユニット上の PORT については、各スレーブユニットの取扱説明書を御覧ください。

◎ C-770AL
◎ CB-08
◎ CDB-5420-AL770
◎ CAD-5410-AL770
◎ C-771
◎ CD-773/ADB5331A
◎ C-772
◎ CB-34/IO

以降、MPL-28-01V1.00/USBW32 は、MPL-28 と称します。

2. 取扱説明書の構成

MPL-28 は、次に示すようにユニットタイプで分類された取扱説明書が用意されていますので、ご使用になるユニットに該当する取扱説明書をご覧ください。

ユニット	ユニット	取扱説明書
I/F ユニット	パソコン-スレーブユニット間のインタフェースユニット ***CB-23/USB***	注.
MCC05 ユニット	パルスジェネレータ MCC05 を搭載したスレーブユニット ***C-770AL　CDB-5420-AL770　CAD-5410-AL770*** ***C-771　CD-773/ADB5331A***	***AL MCC05 ユニット編*** （当取扱説明書）
MCC06 ユニット	パルスジェネレータ MCC06 を搭載したスレーブユニット ***C-772***	***AL MCC06 ユニット編***
I/O ユニット	汎用 I/O ポートを持つスレーブユニット ***CB-08　CB-34/IO***	***AL I/O ユニット編***

注.AL MCC05 ユニット編、AL MCC06 ユニット編、AL I/O ユニット編いずれでも示しています。

3. ご使用になる前に

3-1. DLL の使用上の注意

MPL-28 は、次に示すようにユニットタイプで分類された 4 つの DLL で構成されています。
ユーザアプリケーションは、使用するユニットに該当する DLL の関数を呼び出して、モータコントロール及び、汎用 I/O PORT へのアクセスを行います。よってユーザアプリケーションは、最低でも 2 つの DLL が必要になります。
例.C-772 使用の場合
　AlUsbA.dll、Ac06A.dll が必要になります。
これによりヘッダーファイル(モジュール定義ファイル)、ライブラリーファイルも複数必要になります。

ユニット	ユニット	DLL
I/F ユニット	パソコン-スレーブユニット間のインタフェースユニット *CB-23/USB*	AlUsbA.dll
MCC05 ユニット	パルスジェネレータ MCC05 を搭載したスレーブユニット *C-770AL　CDB-5420-AL770　CAD-5410-AL770* *C-771　CD-773/ADB5331A*	Ac05A.dll
MCC06 ユニット	パルスジェネレータ MCC06 を搭載したスレーブユニット *C-772*	Ac06A.dll
I/O ユニット	汎用 I/O ポートを持つスレーブユニット *CB-08　CB-34/IO*	AcIoA.dll

3-2. この取扱説明書の見方

(1) 各種 DLL の関数と構造体

この取扱説明書の、各種 DLL の構造体、または関数は、次の形式で解説します。

○○○○構造体 ← 構造体の名称　　構造体が使用される関数が存在する DLL の名称 → ○○○○ DLL

○○○○ ← 構造体が使用される関数が対応するマスターボードまたはスレーブユニットの名称

説　明

・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → 構造体の説明

書　式

C 言語 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → C 言語で、構造体を使用するときの定義

VB ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → Visual Basic で、構造体を使用するときの定義

VB.NET ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → Visual Basic.NET で、構造体を使用するときの定義

Delphi ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → Delphi で、構造体を使用するときの定義

メンバ

・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → 構造体のメンバに格納される値の説明

○○○○関数 ← 関数の名称　　関数が存在する DLL の名称 → ○○○○ DLL

○○○○ ← 関数が対応するマスターボードまたはスレーブユニットの名称

機　能

・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → 関数の機能の説明

書　式

C 言語 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → C 言語で、関数を使用するときの定義

VB ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → Visual Basic で、関数を使用するときの定義

VB.NET ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → Visual Basic.NET で、関数を使用するときの定義

Delphi ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → Delphi で、関数を使用するときの定義

引　数

・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → 関数の各引数に指定する値の説明

戻り値

・・・・・・・・・・・・・・・・・・・・・・・・・・・・ → 関数の戻り値の説明

3-3. マルチプロセス対応

複数のユーザアプリケーションで同時に当 DLL を使用することはできません。

3-4. パソコンと I/F ユニット間の通信エラー

USB 通信 I/O エラーが発生した場合、環境設定ツール関数または環境設定直接関数を実行しない限り、エラーはクリアされず、USB 通信は行われません。これらのエラーが発生した場合、再度、環境設定ツール関数または環境設定直接関数を実行してください。

● USB 通信 I/O エラー
- リクエストの最終バイト送信後、200ms 経過しても、アンサーバックの先頭バイトを受信できない
- アンサーバックの受信中に、受信データが得られないまま 200ms 以上経過
- I/F ユニットから受信したデータに異常がある。

【注意】
リトライ回数が 0 以外に設定されいる場合、USB 通信エラー、USB タイムアウトエラー発生後、リトライを行いますが、Windows に割り込まれた場合、リトライ処理に遅れが生じます。

3-5. AL 通信上の通信エラー

AL 通信上の通信エラーが発生した場合、環境設定ツール関数または環境設定直接関数を実行しない限り、エラーはクリアされません。AL 通信上の通信エラーが発生した場合、再度、環境設定ツール関数または環境設定直接関数を実行してください。

※ AL 通信エラーについては、CB-23/USB の取扱説明書をご覧ください。

4. AL I/F ユニット対応 DLL

4-1. 環境設定

AL I/F ユニット対応 DLL では、リトライ回数、AL 通信ボーレートを内部に記憶しています。これらの情報のことを環境設定情報と称します。MPL-28 の各種ライブラリ関数を実行することによる USB 通信、AL シリーズ通信は、環境設定情報をもとに行われるため、ユーザアプリケーションは、次のいずれかの方法により、環境設定情報を設定しなければなりません。

この章では、AL I/F ユニット対応 DLL を、単に I/F DLL と称します。

(1) 環境設定ツールによる設定

環境設定ツールは、画面上から手動操作により環境設定情報を設定するためのツールです。環境設定ツールを使用した場合、環境設定情報に変更があっても、ユーザアプリケーションの修正と再コンパイルが不要になります。環境設定ツールで設定した環境設定情報は、環境設定ツール関数で I/F DLL 内部に取り込まれます。この情報をもとに、USB **通信**のオープン、I/F ユニットの初期化が実行されるため、環境設定ツール関数は、ユーザアプリケーションの先頭で実行してください。

(2) ユーザアプリケーションによる直接設定

ユーザアプリケーションから環境設定直接関数を実行して、環境設定情報を指定します。環境設定直接関数が実行されると、指定された環境設定情報は、I/F DLL 内部に取り込まれます。この情報をもとに、USB 通信のオープン、I/F ユニットの初期化が実行されるため、環境設定直接関数は、ユーザアプリケーションの先頭で実行してください。

4-2. 構造体と関数

● **構造体一覧**

構造体名	説明
RESULT 構造体	関数を実行した結果を格納
環境設定ツール情報構造体	環境設定情報を格納
スレーブ情報構造体	全スレーブのスレーブタイプを格納

● **関数一覧**

関数名	機能
環境設定ツール関数	環境設定ツールで設定した情報で環境設定
環境設定直接関数	設定値を直接指定して環境設定
環境設定ツール情報読み出し関数	環境設定ツールで設定された情報の読み出し
環境設定問い合わせ関数	すでに環境設定が行われているかの問い合わせ
AL 通信エラー累計回数読み出し関数	AL 通信上で発生したエラーの累計回数の読み出し
AL 通信エラー累計回数クリア関数	AL 通信エラー累計回数のクリア
USB 通信エラー累計回数読み出し関数	USB 通信エラーの累計回数の読み出し
USB 通信エラー累計回数クリア関数	USB 通信エラー累計回数のクリア
スレーブ情報読み出し関数	全スレーブのスレーブタイプの読み出し

RESULT 構造体

AL I/F ユニット対応 DLL

CB-23/USB

説　明

I/F DLL の関数を実行した結果が格納されます。

書　式

C 言語
```
typedef struct ALK_TAG_S_RESULT {
    WORD ALK_Result[4];
} ALK_S_RESULT;
```

VB
```
Type ALK_S_RESULT
    ALK_Result(1 To 4) As Integer
End Type
```

VB.NET
```
Structure ALK_S_RESULT
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=4)> Public ALK_Result() As Short
    Public Sub Initialize()
        ReDim ALK_Result(4)
    End Sub
End Structure
```

Delphi
```
ALK_S_RESULT = record
    ALK_Result: array[1..4] of WORD;
end;
```

メンバ

次に示すメンバは、C 言語で表記しています。C 言語の *ALK_Result[0]*～ *ALK_Result[3]*は、Visual Basic では *ALK_Result(1)*～ *ALK_Result(4)*、Visual Basic.NET では *ALK_Result(0)*～ *ALK_Result(3)*、Delphi では *ALK_Result[1]*～ *ALK_Result[4]*に対応します。

ALK_Result[0]　・・・ 実行された関数を示します。このメンバに格納される値は、次のいずれかです。

値	実行された関数
1	環境設定ツール関数
2	環境設定直接関数
100	環境設定ツール情報読み出し関数
101	スレーブ情報読み出し関数

値	実行された関数
102	AL 通信エラー累計回数読み出し関数
103	AL 通信エラー累計回数クリア関数
104	USB 通信エラー累計回数読み出し関数
105	USB 通信エラー累計回数クリア関数

値は 10 進表記です。

ALK_Result[1]　・・・ 実行結果を示します。このメンバに格納される値は、次のいずれかです。

値	実行結果
0	関数の実行が正常に終了しました
1	DLL 内部で API エラーが発生しました
2	NULL ポインタが指定されています
4	指定したパラメータが設定範囲を超えています
8	環境設定されていません
19	I/F ユニットが検出できません
20	USB 通信時に I/O エラーが発生しました
31	DLL が複数のプロセスから同時に使用されました

値は 10 進表記です。

ALK_Result[2]　・・・ I/F ユニット、スレーブユニット上で発生したエラーを示します。
このメンバに格納される値は、次のいずれかです。

値		実行結果
0	(H'00)	エラーは発生しませんでした
1 ～ 127	(H'01 ～ H'7F)	スレーブユニット固有のエラーです。各スレーブの取扱説明書をご覧ください
128	(H'80h)	初期化エラー（スレーブユニットが不正に電源 OFF または RESET されました）
129	(H'81h)	シリアルエラー（スレーブユニットからの受信時にエラーが発生しました）
130	(H'82h)	タイムアウトエラー（スレーブユニットへの送信時にエラーが発生しました）

値は 10 進表記です。()内は、16 進表記です。

ALK_Result[3]　・・・ 将来の拡張用です。

環境設定ツール情報構造体

AL I/F ユニット対応 DLL

CB-23/USB

説　明

環境設定情報が格納されます。

書　式

C 言語
```
typedef struct ALK_TAG_S_ENVIRONMENT_INFO {
    WORD ALK_Ch1;
    WORD ALK_Ch2;
    WORD ALK_Ch3;
    WORD ALK_Retry;
    WORD ALK_RsBaudRate;
    WORD ALK_AlBaudRate;
} ALK_S_ENVIRONMENT_INFO;
```

VB
```
Type ALK_S_ENVIRONMENT_INFO
    ALK_Ch1 As Integer
    ALK_Ch2 As Integer
    ALK_Ch3 As Integer
    ALK_Retry As Integer
    ALK_RsBaudRate As Integer
    ALK_AlBaudRate As Integer
End Type
```

VB.NET
```
Structure ALK_S_ENVIRONMENT_INFO
    Dim ALK_Ch1 As Short
    Dim ALK_Ch2 As Short
    Dim ALK_Ch3 As Short
    Dim ALK_Retry As Short
    Dim ALK_RsBaudRate As Short
    Dim ALK_AlBaudRate As Short
End Structure
```

Delphi
```
ALK_S_ENVIRONMENT_INFO = record
    ALK_Ch1: WORD;
    ALK_Ch2: WORD;
    ALK_Ch3: WORD;
    ALK_Retry: WORD;
    ALK_RsBaudRate: WORD;
    ALK_AlBaudRate: WORD;
end;
```

メンバ

ALK_Ch1　・・・ ALK_USB が格納されます。
ALK_Ch2　・・・ 0 が格納されます。
ALK_Ch3　・・・ 0 が格納されます。
Retry　・・・ 環境設定ツールで設定したリトライ回数が格納されます。0 ～ 3 のいずれかになります。
RsBaudRate　・・・ このメンバに格納される値は、不定となります。
AlBaudRate　・・・ 環境設定ツールで設定した AL 通信ボーレートが格納されます。
このメンバに格納される値は、次のいずれかです。

値（識別子）	値（数値）	AL 通信ボーレート
ALK_RATE_9765	0	9765bps
ALK_RATE_39062	1	39062bps
ALK_RATE_156250	2	156250bps
ALK_RATE_625000	3	625000bps

値（数値）は 10 進表記です。

スレーブ情報構造体

AL I/F ユニット対応 DLL

CB-23/USB

説　明

全スレーブのスレーブタイプを格納します。

書　式

C 言語
```
typedef struct ALK_TAG_S_SLAVE_INFO {
    WORD ALK_SlaveType[31];
} ALK_S_SLAVE_INFO;
```

VB
```
Type ALK_S_SLAVE_INFO
    ALK_SlaveType(1 To 31) As Integer
End Type
```

VB.NET
```
Structure ALK_S_SLAVE_INFO
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=31)> Public ALK_SlaveType() As Short
    Public Sub Initialize()
        ReDim ALK_SlaveType(31)
    End Sub
End Structure
```

Delphi
```
ALK_S_SLAVE_INFO = record
    ALK_SlaveType: array[1..31] of WORD;
end;
```

メンバ

次に示すメンバは、C 言語で表記しています。C 言語の *ALK_SlaveType[0]*～ *ALK_SlaveType[30]*は、Visual Basic では *ALK_SlaveType(1)*～ *ALK_SlaveType(31)*、Visual Basic.NET では *ALK_SlaveType(0)*～ *ALK_SlaveType(30)*、Delphi では *ALK_SlaveType[1]*～ *ALK_SlaveType[31]*に対応します。

ALK_SlaveType[0]　・・・ スレーブアドレス H'01 に接続されているスレーブのスレーブタイプが格納されます。
・・・
ALK_SlaveType[30]　・・・ スレーブアドレス H'1F に接続されているスレーブのスレーブタイプが格納されます。
スレーブが接続されていない場合、これらのメンバには H'FF が格納されます。
スレーブが接続されている場合、これらのメンバに格納される値は、次のいずれかです。

値（識別子）	値（数値）	スレーブタイプ
ALK_SLAVE_C770	H'00	C-770AL
ALK_SLAVE_CB08	H'10	CB-08
ALK_SLAVE_CDB770	H'20	CDB-5420-AL770
ALK_SLAVE_CAD770	H'21	CAD-5410-AL770
ALK_SLAVE_C771	H'01	C-771
ALK_SLAVE_C772	H'02	C-772
ALK_SLAVE_CD773	H'22	CD-773/ADB5331A
ALK_SLAVE_CB34	H'11	CB-34

環境設定ツール関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

環境設定ツールで設定した情報で環境設定を行います。

書　式

C 言語　BOOL ALK_EnvironmentInfo_Tool(ALK_S_RESULT FAR **psResult*);

VB　Function ALK_EnvironmentInfo_Tool(*psResult* As ALK_S_RESULT) As Boolean

VB.NET　Function ALK_EnvironmentInfo_Tool(ByRef *psResult* As ALK_S_RESULT) As Boolean

Delphi　function ALK_EnvironmentInfo_Tool(var *psResult*: ALK_S_RESULT): Boolean;

引　数

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

環境設定直接関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

リトライ回数、USB 通信ボーレート、AL 通信ボーレートを直接指定して環境設定を行います。

書　式

C 言語　BOOL ALK_Environment_Direct(WORD *Retry*, WORD *RsBaudRate*, WORD *AlBaudRate*, ALK_S_RESULT FAR **psResult*);

VB　Function ALK_Environment_Direct(ByVal *Retry* As Integer, ByVal *RsBaudRate* As Integer, ByVal *AlBaudRate* As Integer, *psResult* As ALK_S_RESULT) As Boolean

VB.NET　Function ALK_Environment_Direct(ByVal *Retry* As Short, ByVal *RsBaudRate* As Short, ByVal *AlBaudRate* As Short, ByRef *psResult* As ALK_S_RESULT) As Boolean

Delphi　function ALK_Environment_Direct(*Retry*: WORD; *RsBaudRate*: WORD; *AlBaudRate*: WORD; var *psResult*: ALK_S_RESULT): Boolean;

引　数

Retry　・・・ リトライ回数を指定します。0 ～ 3 のいずれかになります。
RsBaudRate　・・・ 必ず、NULL または 0 を指定します。
AlBaudRate　・・・ AL 通信ボーレートを指定します。

引数 *AlBaudRate* の値	AL 通信ボーレート
ALK_RATE_9765	9765bps
ALK_RATE_39062	39062bps
ALK_RATE_156250	156250bps
ALK_RATE_625000	625000bps

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

環境設定ツール情報読み出し関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

環境設定ツールで設定された情報を読み出します。

書　式

C 言語　BOOL ALK_EnvironmentInfo_Read(ALK_S_ENVIRONMENT_INFO FAR **psEnvInfo*, ALK_S_RESULT FAR **psResult*);

VB　Function ALK_EnvironmentInfo_Read(*psEnvInfo* As ALK_S_ENVIRONMENT_INFO, *psResult* As ALK_S_RESULT) As Boolean

VB.NET　Function ALK_EnvironmentInfo_Read(ByRef *psEnvInfo* As ALK_S_ENVIRONMENT_INFO, ByRef *psResult* As ALK_S_RESULT) As Boolean

Delphi　function ALK_EnvironmentInfo_Read(var *psEnvInfo*: ALK_S_ENVIRONMENT_INFO; var *psResult*: ALK_S_RESULT): Boolean;

引　数

psEnvInfo　・・・ 環境設定ツールで設定された情報が格納される環境設定ツール情報構造体のポインタを指定します。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

関数環境設定問い合わせ関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

すでに環境設定が行われているかを問い合わせます。

書　式

C 言語　BOOL ALK_Environment_Inquiry(VOID);

VB　Function ALK_Environment_Inquiry() As Boolean

VB.NET　Function ALK_Environment_Inquiry() As Boolean

Delphi　function ALK_Environment_Inquiry: Boolean;

引　数

この関数に、引数はありません。

戻り値

すでに環境設定が行われている場合は TRUE(1)、環境設定が行われていない場合は FALSE(0)を返します。

AL 通信エラー累計回数読み出し関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

AL 通信上で発生したエラーの累計回数を読み出します。

書　式

C 言語　BOOL ALK_Err_Count(WORD *IfNo*, WORD FAR **pCount*, ALK_S_RESULT FAR **psResult*);

VB　Function ALK_Err_Count(ByVal *IfNo* As Integer, *pCount* As Integer, *psResult* As ALK_S_RESULT) As Boolean

VB.NET　Function ALK_Err_Count(ByVal *IfNo* As Short, ByRef *pCount* As Short, ByRef *psResult* As ALK_S_RESULT) As Boolean

Delphi　function ALK_Err_Count(*IfNo*: WORD, var *pCount*: WORD, var *psResult*: ALK_S_RESULT): Boolean;

引　数

IfNo　・・・ I/F 番号を指定します。

マスターのボード／ユニット	引数 *IfNo* の値
CB-23/USB	ALK_USB

pCount　・・・ エラーの累計回数が格納される変数のポインタを指定します。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

AL 通信エラー累計回数クリア関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

AL 通信エラー累計回数を 0 にします。

書　式

C 言語　BOOL ALK_Err_Clear(WORD *IfNo*, ALK_S_RESULT FAR **psResult*);

VB　Function ALK_Err_Clear(ByVal *IfNo* As Integer, *psResult* As ALK_S_RESULT) As Boolean

VB.NET　Function ALK_Err_Clear(ByVal *IfNo* As Short, ByRef *psResult* As ALK_S_RESULT) As Boolean

Delphi　function ALK_Err_Clear(*IfNo*: WORD, var *psResult*: ALK_S_RESULT): Boolean;

引　数

IfNo　・・・ I/F 番号を指定します。

マスターのボード／ユニット	引数 *IfNo* の値
CB-23/USB	ALK_USB

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

USB 通信エラー累計回数読み出し関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

環境設定ツール関数、環境設定直接関数実行後、発生した USB 通信エラー累計回数（USB 通信エラーと USB タイムアウトエラーの累計回数）を読み出します。
カウントは最大で 65535 回までカウントし、その後エラーが発生してもカウントはストップします。
カウンタは環境設定ツール関数、環境設定直接関数、USB 通信エラー累計回数クリア関数の実行によって 0 にクリアされます。

書　式

C 言語　BOOL ALK_Usb_Err_Count(WORD FAR *_pCount_, ALK_S_RESULT FAR *_psResult_);

VB　Function ALK_Usb_Err_Count(_pCount_ As Integer, _psResult_ As ALK_S_RESULT) As Boolean

VB.NET　Function ALK_Usb_Err_Count(ByRef _pCount_ As Short, ByRef _psResult_ As ALK_S_RESULT) As Boolean

Delphi　function ALK_Usb_Err_Count(var _pCount_: WORD, var _psResult_: ALK_S_RESULT): Boolean;

引　数

pCount　・・・ エラーの累計回数が格納される変数のポインタを指定します。
psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

USB 通信エラー累計回数クリア関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

USB 通信エラー累計回数を 0 にします。

書　式

C 言語　BOOL ALK_Usb_Err_Clear(ALK_S_RESULT FAR *_psResult_);

VB　Function ALK_Usb_Err_Clear(_psResult_ As ALK_S_RESULT) As Boolean

VB.NET　Function ALK_Usb_Err_Clear(ByRef _psResult_ As ALK_S_RESULT) As Boolean

Delphi　function ALK_Usb_Err_Clear(var _psResult_: ALK_S_RESULT): Boolean;

引　数

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

スレーブ情報読み出し関数

AL I/F ユニット対応 DLL

CB-23/USB

機　能

スレーブアドレス H'01 ～ H'1F に接続されているスレーブのスレーブタイプを読み出します。

書　式

C 言語　BOOL ALK_SlaveInfo_Read(WORD *IfNo*, ALK_S_SLAVE_INFO FAR **psSlvInfo*, ALK_S_RESULT FAR **psResult*);

VB　Function ALK_SlaveInfo_Read(ByVal *IfNo* As Integer, *psSlvInfo* As ALK_S_SLAVE_INFO, *psResult* As ALK_S_RESULT) As Boolean

VB.NET　Function ALK_SlaveInfo_Read(ByVal *IfNo* As Short, ByRef *psSlvInfo* As ALK_S_SLAVE_INFO, ByRef *psResult* As ALK_S_RESULT) As Boolean

Delphi　function ALK_SlaveInfo_Read(*IfNo*: WORD; var *psSlvInfo*: ALK_S_SLAVE_INFO; var *psResult*: ALK_S_RESULT): Boolean;

引　数

IfNo　・・・ I/F 番号を指定します。

マスターのボード／ユニット	引数 *IfNo* の値
CB-23/USB	ALK_USB

psSlvInfo　・・・ スレーブ情報が格納されるスレーブ情報構造体のポインタを指定します。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

5. AL MCC05 ユニット対応 DLL

AL MCC05 ユニット対応 DLL は、パルスジェネレータ MCC05 を搭載したスレーブユニット上の、MCC05、汎用 I/O PORT、モータ制御用 I/O を制御するための各種関数を備えています。

この章では、AL MCC05 ユニット対応 DLL を、単に MCC05 DLL と称します。

5-1. デバイスの制御

MCC05 の 1 軸分を、デバイスと称します。
デバイスを制御するための説明をします。

(1) デバイス上の PORT の説明

MCC05 DLL は、デバイス上の PORT を直接制御します。

スレーブユニット	デバイス上の PORT
C-770AL, CDB-5420-AL770, CAD-5410-AL770, C-771, CD-773/ADB5331A	DRIVE COMMAND PORT DRIVE DATA1 PORT DRIVE DATA2 PORT DRIVE DATA3 PORT STATUS1 PORT STATUS2 PORT STATUS3 PORT STATUS4 PORT STATUS5 PORT COUNTER COMMAND PORT COUNTER DATA1 PORT COUNTER DATA2 PORT COUNTER DATA3 PORT

(2) デバイスのオープンとクローズ

デバイスを制御する前に、デバイスをオープンし、デバイスハンドルを取得します。以降、デバイスを制御する関数を実行する際に、このデバイスハンドルを指定します。
デバイスハンドルは、デバイスをクローズするまで有効です。ユーザアプリケーション終了時は、必ずデバイスをクローズしてください。クローズが行われていないと、以降正常に動作しません。

5-2. 汎用 I/O PORT の制御

スレーブユニット上の汎用 I/O PORT を制御するための説明をします。

(1) 汎用 I/O PORT の説明

MCC05 DLL は、汎用 I/O PORT を直接制御します。

スレーブユニット	汎用 I/O PORT
C-770AL	あり
CDB-5420-AL770	なし
CAD-5410-AL770	なし
C-771	なし
CD-773/ADB5331A	なし

(2) 汎用 I/O PORT のオープンとクローズ

I/O PORT を制御する前に、I/O PORT をオープンし、I/O PORT ハンドルを取得します。以降、I/O PORT を制御する関数を実行する際に、この I/O PORT ハンドルを指定します。
I/O PORT ハンドルは、I/O PORT をクローズするまで有効です。ユーザアプリケーション終了時は、必ず I/O PORT をクローズしてください。クローズが行われていないと、以降正常に動作しません。

5-3. モータ制御用 I/O の制御

スレーブユニット上のモータ制御用 I/O を制御するための説明をします。

(1) モータ制御用 I/O の説明

MCC05 DLL は、モータ制御用 I/O を直接制御します。

スレーブユニット	モータ制御用 I/O
C-770AL	なし
CDB-5420-AL770	制御信号
CAD-5410-AL770	制御信号
C-771	制御 I/O PORT
CD-773/ADB5331A	制御 I/O PORT

(2) 制御信号または制御 I/O PORT のオープンとクローズ

モータ制御用 I/O を制御する前に、制御信号または制御 I/O PORT をオープンし、制御信号ハンドルを取得します。以降、制御信号または制御 I/O PORT を制御する関数を実行する際に、この制御信号ハンドルを指定します。
制御信号ハンドルは、制御信号または制御 I/O PORT をクローズするまで有効です。ユーザアプリケーション終了時は、必ず制御信号または制御 I/O PORT をクローズしてください。クローズが行われていないと、以降正常に動作しません。

5-4. 構造体と関数

● 構造体一覧

構造体名	説明
RESULT 構造体	関数を実行した結果を格納
データ構造体	デバイスに一括でアクセスするためのデータを格納

● 関数一覧

関数名	機能
デバイスオープン関数	デバイスのオープン
デバイスクローズ関数	デバイスのクローズ
DRIVE COMMAND 一括書き込み関数	DRIVE COMMAND, DATA1~3 PORT に一括書き込み
DRIVE DATA PORT 一括書き込み関数	DRIVE DATA1~3 PORT に一括書き込み
DRIVE COMMAND PORT 書き込み関数	DRIVE COMMAND PORT に書き込み
DRIVE DATA1 PORT 書き込み関数	DRIVE DATA1 PORT に書き込み
DRIVE DATA2 PORT 書き込み関数	DRIVE DATA2 PORT に書き込み
DRIVE DATA3 PORT 書き込み関数	DRIVE DATA3 PORT に書き込み
STATUS1 PORT 読み出し関数	STATUS1 PORT の読み出し
STATUS2 PORT 読み出し関数	STATUS2 PORT の読み出し
STATUS3 PORT 読み出し関数	STATUS3 PORT の読み出し
STATUS4 PORT 読み出し関数	STATUS4 PORT の読み出し
STATUS5 PORT 読み出し関数	STATUS5 PORT の読み出し
DRIVE DATA PORT 一括読み出し関数	DRIVE DATA1~3 PORT の一括読み出し
DRIVE DATA1 PORT 読み出し関数	DRIVE DATA1 PORT の読み出し
DRIVE DATA2 PORT 読み出し関数	DRIVE DATA2 PORT の読み出し
DRIVE DATA3 PORT 読み出し関数	DRIVE DATA3 PORT の読み出し
READY WAIT 関数	READY になるまで待機
READY WAIT 状態読み出し関数	READY WAIT 関数の状態の読み出し
READY WAIT 中止関数	READY WAIT 関数の READY 待ちを中止
COUNTER COMMAND 一括書き込み関数	COUNTER COMMAND, DATA1~3 PORT に一括書き込み
COUNTER COMMAND PORT 書き込み関数	COUNTER COMMAND PORT に書き込み
COUNTER DATA1 PORT 書き込み関数	COUNTER DATA1 PORT に書き込み
COUNTER DATA2 PORT 書き込み関数	COUNTER DATA2 PORT に書き込み
COUNTER DATA3 PORT 書き込み関数	COUNTER DATA3 PORT に書き込み
汎用 I/O PORT オープン関数	汎用 I/O PORT のオープン
汎用 I/O PORT クローズ関数	汎用 I/O PORT のクローズ
汎用 I/O PORT 一括読み出し関数	汎用 I/O PORT の状態を一括読み出し
汎用 I/O PORT 指定信号読み出し関数	汎用 I/O PORT の指定信号の状態の読み出し
汎用 I/O PORT 一括書き込み関数	汎用 I/O PORT に一括書き込み
汎用 I/O PORT 指定信号 LOW レベル書き込み関数	汎用 I/O PORT の指定信号に LOW レベルを書き込み
汎用 I/O PORT 指定信号 HIGH レベル書き込み関数	汎用 I/O PORT の指定信号に HIGH レベルを書き込み
制御信号オープン関数	制御信号のオープン
制御信号クローズ関数	制御信号のクローズ
制御信号読み出し関数	制御信号の状態を読み出し
制御信号指定ビット読み出し関数	制御信号の指定ビットの状態を読み出し
制御信号書き込み関数	制御信号に書き込み
制御信号指定ビット ON 書き込み関数	制御信号の指定ビットに ON を書き込み
制御信号指定ビット OFF 書き込み関数	制御信号の指定ビットに OFF を書き込み
データセット関数	24 ビットデータをデータ構造体に格納
データゲット関数	データ構造体のデータを 24 ビットデータに変換

RESULT **構造体**

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

説　明

MCC05 DLL 関数を実行した結果が格納されます。

書　式

C 言語
```
typedef struct AC05_TAG_S_RESULT {
    WORD AC05_Result[4];
} AC05_S_RESULT;
```

VB
```
Type AC05_S_RESULT
    AC05_Result(1 To 4) As Interger
End Type
```

VB.NET
```
Structure AC05_S_RESULT
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=4)> Public AC05_Result() As Short
    Public Sub Initialize()
        ReDim AC05_Result(4)
    End Sub
End Structure
```

Delphi
```
AC05_S_RESULT = record
    AC05_Result: array[1..4] of WORD;
end;
```

メンバ

次に示すメンバは、C 言語で表記しています。C 言語の *AC05_Result[0]*～ *AC05_Result[3]*は、Visual Basic では *AC05_Result(1)*～ *AC05_Result(4)*、Visual Basic.NET では *AC05_Result(0)*～ *AC05_Result(3)*、Delphi では *AC05_Result[1]*～ *AC05_Result[4]*に対応します。

AC05_Result[0]　･･･ 実行された関数を示します。このメンバに格納される値は、次のいずれかです。

値	実行された関数
10	デバイスオープン関数
11	デバイスクローズ関数
20	DRIVE COMMAND PORT 書き込み関数
21	DRIVE DATA1 PORT 書き込み関数
22	DRIVE DATA2 PORT 書き込み関数
23	DRIVE DATA3 PORT 書き込み関数
30	COUNTER COMMAND PORT 書き込み関数
31	COUNTER DATA1 PORT 書き込み関数
32	COUNTER DATA2 PORT 書き込み関数
33	COUNTER DATA3 PORT 書き込み関数
41	STATUS1 PORT 読み出し関数
42	STATUS2 PORT 読み出し関数
43	STATUS3 PORT 読み出し関数
44	STATUS4 PORT 読み出し関数
45	STATUS5 PORT 読み出し関数
51	DRIVE DATA1 PORT 読み出し関数
52	DRIVE DATA2 PORT 読み出し関数
53	DRIVE DATA3 PORT 読み出し関数
60	汎用 I/O PORT オープン関数
61	汎用 I/O PORT クローズ関数
62	汎用 I/O PORT 一括読み出し関数
63	汎用 I/O PORT 指定信号読み出し関数

値	実行された関数
64	汎用 I/O PORT 一括書き込み関数
67	汎用 I/O PORT 指定信号 LOW レベル書き込み関数
68	汎用 I/O PORT 指定信号 HIGH レベル書き込み関数
70	制御信号オープン関数
71	制御信号クローズ関数
72	制御信号読み出し関数
73	制御信号書き込み関数
74	制御信号指定ビット読み出し関数
75	制御信号指定ビット ON 書き込み関数
76	制御信号指定ビット OFF 書き込み関数
80	READY WAIT 関数
81	READY WAIT 状態読み出し関数
82	READY WAIT 中止関数
90	DRIVE COMMAND 一括書き込み関数
91	DRIVE DATA PORT 一括読み出し関数
92	COUNTER COMMAND 一括書き込み関数
93	DRIVE DATA PORT 一括書き込み関数

値は 10 進表記です。

AC05_Result[1]　・・・ 実行結果を示します。このメンバに格納される値は、次のいずれかです。

値	実行結果
0	関数の実行が正常に終了しました
1	DLL 内部で API エラーが発生しました
2	NULL ポインタが指定されています
4	指定したパラメータが設定範囲を超えています
6	ハンドルの値が範囲外です
8	環境設定されていません
9	指定されたアドレスにスレーブが接続されていません
10	指定したスレーブタイプと実際に接続されているスレーブのタイプが異なります
11	接続されているスレーブでは、関数の実行ができません
12	指定されたデバイス、汎用 I/O PORT、制御信号（制御 I/O PORT）は、オープンされていません
13	指定されたデバイス、汎用 I/O PORT、制御信号（制御 I/O PORT）は、すでにオープンされています
14	READY WAIT 関数が TIME OVER で終了しています
15	WM_QUIT メッセージを受信しました
16	READY WAIT 中に READY WAIT 中止関数が実行されました
17	同一デバイスの READY WAIT 関数が複数同時に実行されました
18	デバイス又は、I/O PORT 数が最大値を越えています
29	現在のスレッドからは、操作出来ません
30	I/F ユニット DLL が見つかりません

値は 10 進表記です。

AC05_Result[2]　・・・ I/F ユニット、スレーブユニット上で発生したエラーを示します。
このメンバに格納される値は、次のいずれかです。

値		実行結果
0	(H'00)	エラーは発生しませんでした
1 ～ 127	(H'01 ～ H'7F)	スレーブユニット固有のエラーです。各スレーブの取扱説明書をご覧ください
128	(H'80h)	初期化エラー（スレーブユニットが不正に電源 OFF または RESET されました）
129	(H'81h)	シリアルエラー（スレーブユニットからの受信時にエラーが発生しました）
130	(H'82h)	タイムアウトエラー（スレーブユニットへの送信時にエラーが発生しました）

値は 10 進表記です。()内は、16 進表記です。

AC05_Result[3]　・・・ 将来の拡張用です。

データ構造体

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

説　明

データを一括で読み書きするときに使用します。

● データを一括で読み書きするとき
- DRIVE DATA1 PORT、DRIVE DATA2 PORT、DRIVE DATA3 PORT のデータを一括で書き込むとき
- DRIVE DATA1 PORT、DRIVE DATA2 PORT、DRIVE DATA3 PORT のデータを一括で読み出すとき
- COUNTER DATA1 PORT、COUNTER DATA2 PORT、COUNTER DATA3 PORT のデータを一括で書き込むとき

書　式

C 言語
```
typedef struct AC05_TAG_S_DATA {
    WORD AC05_Data[4];
} AC05_S_DATA;
```

VB
```
Type AC05_S_DATA
    AC05_Data(1 To 4) As Integer
End Type
```

VB.NET
```
Structure AC05_S_DATA
    <MarshalAs(UnmanagedType.ByValArray, SizeConst:=4)> Public AC05_Data() As Short
    Public Sub Initialize()
        ReDim AC05_Data(4)
    End Sub
End Structure
```

Delphi
```
AC05_S_DATA = record
    AC05_Data: array[1..4] of WORD;
end;
```

メンバ

次に示すメンバは、C 言語で表記しています。C 言語の *AC05_Data[0]*～ *AC05_Data[3]*は、Visual Basic では *AC05_Data(1)*～ *AC05_Data(4)*、Visual Basic.NET では *AC05_Data(0)*～ *AC05_Data(3)*、Delphi では *AC05_Data[1]*～ *AC05_Data[4]*に対応します。

AC05_Data[0]　・・・ DRIVE DATA1 PORT、COUNTER DATA1 PORT のいずれかの内容を格納します。
AC05_Data[1]　・・・ DRIVE DATA2 PORT、COUNTER DATA2 PORT のいずれかの内容を格納します。
AC05_Data[2]　・・・ DRIVE DATA3 PORT、COUNTER DATA3 PORT のいずれかの内容を格納します。
AC05_Data[3]　・・・ 将来の拡張用です。

デバイスオープン関数

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

I/F 番号、スレーブアドレス、軸、スレーブタイプを指定して、デバイスをオープンし、引数 *phDev* で示される変数にデバイスハンドルを格納します。

書　式

C 言語 BOOL AC05_BOpen(WORD *IfNo*, WORD *SlaveAddr*, WORD *Axis*, WORD *SlaveType*, DWORD FAR **phDev*, AC05_S_RESULT FAR **psResult*);

VB Function AC05_BOpen(ByVal *IfNo* As Integer, ByVal *SlaveAddr* As Integer, ByVal *Axis* As Integer, ByVal *SlaveType* As Integer, *phDev* As Long, *psResult* As AC05_S_RESULT) As Boolean

VB.NET Function AC05_BOpen(ByVal *IfNo* As Short, ByVal *SlaveAddr* As Short, ByVal *Axis* As Short, ByVal *SlaveType* As Short, ByRef *phDev* As Integer, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi function AC05_BOpen(*IfNo*: WORD; *SlaveAddr*: WORD; *Axis*: WORD; *SlaveType*: WORD; var *phDev*: DWORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

IfNo ･･･ I/F 番号を指定します。

マスターのボード／ユニット	引数 *IfNo* の値
CB-09	ボード番号（0 ～ 9）
CB-23/USB	AC05_USB

SlaveAddr ･･･ スレーブアドレスを指定します。H'01 ～ H'1F のいずれかになります。

Axis ･･･ 軸を指定します。この引数は、次の値のいずれかになります。

● スレーブタイプが C-770AL、C-771、CD-773/ADB5331A の場合

引数 *Axis* の値	軸	引数 *Axis* の値	軸
AC05_X	X 軸	AC05_Z	Z 軸
AC05_Y	Y 軸	AC05_A	A 軸

● スレーブタイプが CDB-5420-AL770、CAD-5410-AL770 の場合
何を指定しても無視されます

SlaveType ･･･ スレーブタイプを指定します。

引数 *SlaveType* の値	スレーブタイプ
AC05_SLAVE_C770	C-770AL
AC05_SLAVE_CDB770	CDB-5420-AL770
AC05_SLAVE_CAD770	CAD-5410-AL770
AC05_SLAVE_C771	C-771
AC05_SLAVE_CD773	CD-773/ADB5331A

phDev ･･･ デバイスハンドルが格納される変数のポインタを指定します。

psResult ･･･ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

デバイスクローズ関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスをクローズします。

書　式

C 言語　BOOL AC05_BClose(DWORD *hDev*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BClose(ByVal *hDev* As Long, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BClose(ByVal *hDev* As Integer, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BClose(*hDev*: DWORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。
psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

DRIVE COMMAND 一括書き込み関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの DRIVE COMMAND PORT、DRIVE DATA1 PORT、DRIVE DATA2 PORT、DRIVE DATA3 PORT にコマンドコード、データを一括書き込みします。

書　式

C 言語　BOOL AC05_IWDrive(DWORD *hDev*, WORD Cmd, AC05_S_DATA FAR **psData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_IWDrive(ByVal *hDev* As Long, ByVal *Cmd* As Integer, *psData* As AC05_S_DATA, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_IWDrive(ByVal *hDev* As Integer, ByVal *Cmd* As Short, ByRef *psData* As AC05_S_DATA, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_IWDrive(*hDev*: DWORD; *Cmd*: WORD; var *psData*: AC05_S_DATA; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。
Cmd　・・・ 書き込むコマンドコードを指定します。下位 8 ビットが有効です。
psData　・・・ 書き込むデータが格納されているデータ構造体のポインタを指定します。
データ構造体の各メンバは、下位 8 ビットが有効です。
psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

DRIVE DATA PORT **一括書き込み関数**

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの DRIVE DATA1 PORT、DRIVE DATA2 PORT、DRIVE DATA3 PORT にデータを一括書き込みします。

書　式

C 言語 BOOL AC05_IWData(DWORD *hDev*, AC05_S_DATA FAR **psData*, AC05_S_RESULT FAR **psResult*);

VB Function AC05_IWData(ByVal *hDev* As Long, *psData* As AC05_S_DATA, *psResult* As AC05_S_RESULT) As Boolean

VB.NET Function AC05_IWData(ByVal *hDev* As Integer, ByRef *psData* As AC05_S_DATA, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi function AC05_IWData(*hDev*: DWORD; var *psData*: AC05_S_DATA; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev ・・・ デバイスハンドルを指定します。
psData ・・・ 書き込むデータが格納されているデータ構造体のポインタを指定します。
データ構造体の各メンバは、下位 8 ビットが有効です。
psResult ・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

DRIVE COMMAND PORT **書き込み関数**

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの DRIVE COMMAND PORT にコマンドコードを書き込みます。

書　式

C 言語 BOOL AC05_BWDriveCommand(DWORD *hDev*, WORD FAR **pCmd*, AC05_S_RESULT FAR **psResult*);

VB Function AC05_BWDriveCommand(ByVal *hDev* As Long, *pCmd* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET Function AC05_BWDriveCommand(ByVal *hDev* As Integer, ByRef *pCmd* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi function AC05_BWDriveCommand(*hDev*: DWORD; var *pCmd*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev ・・・ デバイスハンドルを指定します。
pCmd ・・・ 書き込むコマンドコードが格納されている変数のポインタを指定します。
書き込むコマンドコードが格納されている変数は、下位 8 ビットが有効です。
psResult ・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

DRIVE DATA1 PORT **書き込み関数**

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの DRIVE DATA1 PORT にデータを書き込みます。

書　式

C 言語　BOOL AC05_BWDriveData1(DWORD *hDev*, WORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BWDriveData1(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BWDriveData1(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BWDriveData1(*hDev*: DWORD; var *pData*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。

pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
書き込むデータが格納されている変数は、下位 8 ビットが有効です。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

DRIVE DATA2 PORT **書き込み関数**

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの DRIVE DATA2 PORT にデータを書き込みます。

書　式

C 言語　BOOL AC05_BWDriveData2(DWORD *hDev*, WORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BWDriveData2(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BWDriveData2(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BWDriveData2(*hDev*: DWORD; var *pData*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。

pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
書き込むデータが格納されている変数は、下位 8 ビットが有効です。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

DRIVE DATA3 PORT **書き込み関数**

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの DRIVE DATA3 PORT にデータを書き込みます。

書　式

C 言語　BOOL AC05_BWDriveData3(DWORD *hDev*, WORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BWDriveData3(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BWDriveData3(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BWDriveData3(*hDev*: DWORD; var *pData*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。

pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
書き込むデータが格納されている変数は、下位 8 ビットが有効です。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

STATUS1 PORT **読み出し関数**

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの STATUS1 PORT を読み出します。

書　式

C 言語　BOOL AC05_BRStatus1(DWORD *hDev*, WORD FAR **pStatus*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BRStatus1(ByVal *hDev* As Long, *pStatus* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BRStatus1(ByVal *hDev* As Integer, ByRef *pStatus* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BRStatus1(*hDev*: DWORD; var *pStatus*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。

pStatus　・・・ 読み出した内容が格納される変数のポインタを指定します。
読み出した内容は、下位 8 ビットが有効です。上位 8 ビットには、0 が読み出されます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

STATUS2 PORT 読み出し関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの STATUS2 PORT を読み出します。

書　式

C 言語　BOOL AC05_BRStatus2(DWORD *hDev*, WORD FAR **pStatus*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BRStatus2(ByVal *hDev* As Long, *pStatus* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BRStatus2(ByVal *hDev* As Integer, ByRef *pStatus* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BRStatus2(*hDev*: DWORD; var *pStatus*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。

pStatus　・・・ 読み出した内容が格納される変数のポインタを指定します。
読み出した内容は、下位 8 ビットが有効です。上位 8 ビットには、0 が読み出されます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

STATUS3 PORT 読み出し関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの STATUS3 PORT を読み出します。

書　式

C 言語　BOOL AC05_BRStatus3(DWORD *hDev*, WORD FAR **pStatus*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BRStatus3(ByVal *hDev* As Long, *pStatus* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BRStatus3(ByVal *hDev* As Integer, ByRef *pStatus* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BRStatus3(*hDev*: DWORD; var *pStatus*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。

pStatus　・・・ 読み出した内容が格納される変数のポインタを指定します。
読み出した内容は、下位 8 ビットが有効です。上位 8 ビットには、0 が読み出されます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

STATUS4 PORT 読み出し関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの STATUS4 PORT を読み出します。

書　式

C 言語　BOOL AC05_BRStatus4(DWORD *hDev*, WORD FAR **pStatus*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BRStatus4(ByVal *hDev* As Long, *pStatus* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BRStatus4(ByVal *hDev* As Integer, ByRef *pStatus* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BRStatus4(*hDev*: DWORD; var *pStatus*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。

pStatus　・・・ 読み出した内容が格納される変数のポインタを指定します。
読み出した内容は、下位 8 ビットが有効です。上位 8 ビットには、0 が読み出されます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

STATUS5 PORT 読み出し関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの STATUS5 PORT を読み出します。

書　式

C 言語　BOOL AC05_BRStatus5(DWORD *hDev*, WORD FAR **pStatus*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BRStatus5(ByVal *hDev* As Long, *pStatus* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BRStatus5(ByVal *hDev* As Integer, ByRef *pStatus* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BRStatus5(*hDev*: DWORD; var *pStatus*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。

pStatus　・・・ 読み出した内容が格納される変数のポインタを指定します。
読み出した内容は、下位 8 ビットが有効です。上位 8 ビットには、0 が読み出されます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

DRIVE DATA PORT **一括読み出し関数**

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの DRIVE DATA1 PORT、DRIVE DATA2 PORT、DRIVE DATA3 PORT を一括読み出しします。

書　式

C 言語 BOOL AC05_IRDrive(DWORD *hDev*, AC05_S_DATA FAR **psData*, AC05_S_RESULT FAR **psResult*);

VB Function AC05_IRDrive(ByVal *hDev* As Long, *psData* As AC05_S_DATA, *psResult* As AC05_S_RESULT) As Boolean

VB.NET Function AC05_IRDrive(ByVal *hDev* As Integer, ByRef *psData* As AC05_S_DATA, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi function AC05_IRDrive(*hDev*: DWORD; var *psData*: AC05_S_DATA; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev ・・・ デバイスハンドルを指定します。

psData ・・・ 読み出したデータが格納されるデータ構造体のポインタを指定します。
データ構造体の各メンバは、下位 8 ビットが有効です。上位 8 ビットには 0 が格納されます。

psResult ・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

DRIVE DATA1 PORT **読み出し関数**

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの DRIVE DATA1 PORT を読み出します。

書　式

C 言語 BOOL AC05_BRDriveData1(DWORD *hDev*, WORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB Function AC05_BRDriveData1(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET Function AC05_BRDriveData1(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi function AC05_BRDriveData1(*hDev*: DWORD; var *pData*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev ・・・ デバイスハンドルを指定します。

pData ・・・ 読み出した内容が格納される変数のポインタを指定します。
読み出した内容は、下位 8 ビットが有効です。上位 8 ビットには、0 が読み出されます。

psResult ・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

DRIVE DATA2 PORT 読み出し関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの DRIVE DATA2 PORT を読み出します。

書　式

C 言語　BOOL AC05_BRDriveData2(DWORD *hDev*, WORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BRDriveData2(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BRDriveData2(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC05_S_RESULT)As Boolean

Delphi　function AC05_BRDriveData2(*hDev*: DWORD; var *pData*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。

pData　・・・ 読み出した内容が格納される変数のポインタを指定します。
読み出した内容は、下位 8 ビットが有効です。上位 8 ビットには、0 が読み出されます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

DRIVE DATA3 PORT 読み出し関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの DRIVE DATA3 PORT を読み出します。

書　式

C 言語　BOOL AC05_BRDriveData3(DWORD *hDev*, WORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BRDriveData3(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BRDriveData3(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC05_S_RESULT)As Boolean

Delphi　function AC05_BRDriveData3(*hDev*: DWORD; var *pData*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。

pData　・・・ 読み出した内容が格納される変数のポインタを指定します。
読み出した内容は、下位 8 ビットが有効です。上位 8 ビットには、0 が読み出されます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

READY WAIT **関数**

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスが READY（STATUS1 PORT の BUSY BIT = 0）になるまで待機します。最大待ち時間を越えるとエラー終了します。

書　式

C 言語 BOOL AC05_BWaitDriveCommand(DWORD *hDev*, WORD *WaitTime*, AC05_S_RESULT FAR **psResult*);

VB Function AC05_BWaitDriveCommand(ByVal *hDev* As Long, ByVal *WaitTime* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET Function AC05_BWaitDriveCommand(ByVal *hDev* As Integer, ByVal *WaitTime* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi function AC05_BWaitDriveCommand(*hDev*: DWORD; *WaitTime*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev ・・・ デバイスハンドルを指定します。
WaitTime ・・・ 最大待ち時間を 1ms 単位で設定します。0 を指定すると READY になるまで無限に待機します。
psResult ・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

READY WAIT **状態読み出し関数**

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの READY WAIT 関数の状態を返します。

書　式

C 言語 BOOL AC05_BIsWait(DWORD *hDev*, WORD FAR **pWaitSts*, AC05_S_RESULT FAR **psResult*);

VB Function AC05_BIsWait(ByVal *hDev* As Long, *pWaitSts* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET Function AC05_BIsWait(ByVal *hDev* As Integer, ByRef *pWaitSts* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi function AC05_BIsWait(*hDev*: DWORD; var *pWaitSts*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev ・・・ デバイスハンドルを指定します。
pWaitSts ・・・ READY WAIT 関数の状態が格納される変数のポインタを指定します。

格納される値	READY WAIT 関数の状態
0	READY になるまで待機していない状態
1	READY になるまで待機している状態

psResult ・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

READY WAIT 中止関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの READY WAIT 関数の READY 待ちを中止します。

書　式

C 言語　BOOL AC05_BBreakWait(DWORD *hDev*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BBreakWait(ByVal *hDev* As Long, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BBreakWait(ByVal *hDev* As Integer, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BBreakWait(*hDev*: DWORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。
psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

COUNTER COMMAND 一括書き込み関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの COUNTER COMMAND PORT、COUNTER DATA1 PORT、COUNTER DATA2 PORT、COUNTER DATA3 PORT にコマンドコード、データを一括書き込みします。

書　式

C 言語　BOOL AC05_IWCounter(DWORD *hDev*, WORD *Cmd*, AC05_S_DATA FAR **psData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_IWCounter(ByVal *hDev* As Long, ByVal *Cmd* As Integer, *psData* As AC05_S_DATA, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_IWCounter(ByVal *hDev* As Integer, ByVal *Cmd* As Short, ByRef *psData* As AC05_S_DATA, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_IWCounter(*hDev*: DWORD; *Cmd*: WORD; var *psData*: AC05_S_DATA; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。
Cmd　・・・ 書き込むコマンドコードを指定します。下位 8 ビットが有効です。
psData　・・・ 書き込むデータが格納されているデータ構造体のポインタを指定します。
データ構造体の各メンバは、下位 8 ビットが有効です。
psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

COUNTER COMMAND PORT 書き込み関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの COUNTER COMMAND PORT にコマンドを書き込みます。

書　式

C 言語　BOOL AC05_BWCounterCommand(DWORD *hDev*, WORD FAR **pCmd*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BWCounterCommand(ByVal *hDev* As Long, *pCmd* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BWCounterCommand(ByVal *hDev* As Integer, ByRef *pCmd* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BWCounterCommand(*hDev*: DWORD; var *pCmd*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。
pCmd　・・・ 書き込むコマンドコードが格納されている変数のポインタを指定します。
書き込むコマンドコードが格納されている変数は、下位 8 ビットが有効です。
psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

COUNTER DATA1 PORT 書き込み関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの COUNTER DATA1 PORT にデータを書き込みます。

書　式

C 言語　BOOL AC05_BWCounterData1(DWORD *hDev*, WORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BWCounterData1(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BWCounterData1(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BWCounterData1(*hDev*: DWORD; var *pData*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。
pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
書き込むデータが格納されている変数は、下位 8 ビットが有効です。
psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

COUNTER DATA2 PORT 書き込み関数

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの COUNTER DATA2 PORT にデータを書き込みます。

書　式

C 言語　BOOL AC05_BWCounterData2(DWORD *hDev*, WORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BWCounterData2(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BWCounterData2(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BWCounterData2(*hDev*: DWORD; var *pData*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。

pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
書き込むデータが格納されている変数は、下位 8 ビットが有効です。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

COUNTER DATA3 PORT 書き込み関数

AL MCC05 **ユニット対応** DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定されたデバイスの COUNTER DATA3 PORT にデータを書き込みます。

書　式

C 言語　BOOL AC05_BWCounterData3(DWORD *hDev*, WORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BWCounterData3(ByVal *hDev* As Long, *pData* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BWCounterData3(ByVal *hDev* As Integer, ByRef *pData* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BWCounterData3(*hDev*: DWORD; var *pData*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hDev　・・・ デバイスハンドルを指定します。

pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
書き込むデータが格納されている変数は、下位 8 ビットが有効です。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT オープン関数

AL MCC05 ユニット対応 DLL

C-770AL

機　能

I/F 番号、スレーブアドレス、スレーブタイプを指定して、汎用 I/O PORT をオープンし、引数 *phPort* で示される変数に汎用 I/O PORT ハンドルを格納します。

書　式

C 言語　BOOL AC05_BPortOpen(WORD *IfNo*, WORD *SlaveAddr*, WORD *SlaveType*, DWORD FAR **phPort*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BPortOpen(ByVal *IfNo* As Integer, ByVal *SlaveAddr* As Integer, ByVal *SlaveType* As Integer, *phPort* As Long, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BPortOpen(ByVal *IfNo* As Short, ByVal *SlaveAddr* As Short, ByVal *SlaveType* As Short, ByRef *phPort* As Integer, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BPortOpen(*IfNo*: WORD; *SlaveAddr*: WORD; *SlaveType*: WORD; var *phPort*: DWORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

IfNo　・・・ I/F 番号を指定します。

マスターのボード／ユニット	引数 *IfNo* の値
CB-09	ボード番号（0 ～ 9）
CB-23/USB	AC05_USB

SlaveAddr　・・・ スレーブアドレスを指定します。H'01 ～ H'1F のいずれかになります。

SlaveType　・・・ スレーブタイプを指定します。

引数 *SlaveType* の値	スレーブタイプ
AC05_SLAVE_C770	C-770AL

phPort　・・・ 汎用 I/O PORT ハンドルが格納される変数のポインタを指定します。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT クローズ関数

AL MCC05 ユニット対応 DLL

C-770AL

機　能

指定された汎用 I/O PORT をクローズします。

書　式

C 言語　BOOL AC05_BPortClose(DWORD *hPort*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BPortClose(ByVal *hPort* As Long, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BPortClose(ByVal *hPort* As Integer, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BPortClose(*hPort*: DWORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hPort　・・・ 汎用 I/O PORT ハンドルを指定します。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT 一括読み出し関数

AL MCC05 ユニット対応 DLL

C-770AL

機　能

汎用 I/O PORT の状態を一括読み出しします。

書　式

C 言語　BOOL AC05_BPortIn(DWORD *hPort*, DWORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BPortIn(ByVal *hPort* As Long, *pData* As Long, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BPortIn(ByVal *hPort* As Integer, ByRef *pData* As Integer, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BPortIn(*hPort*: DWORD; var *pData*: DWORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hPort　・・・ 汎用 I/O PORT ハンドルを指定します。

pData　・・・ 読み出した内容が格納される変数のポインタを指定します。
変数の内容は、次のようになります。

● C-770AL の汎用 I/O PORT を読み出す場合

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	$\overline{IN1}$	$\overline{IN0}$	0	0	$\overline{OUT1}$	$\overline{OUT0}$

HIGH レベルのときに 0、LOW レベルのときに 1 が読み出されます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT 指定信号読み出し関数

AL MCC05 ユニット対応 DLL

C-770AL

機　能

汎用 I/O PORT の指定信号の状態を読み出します。

書　式

C 言語　BOOL AC05_BSigIn(DWORD *hPort*, WORD *SigNo*, WORD FAR **pStatus*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BSigIn(ByVal *hPort* As Long, ByVal *SigNo* As Integer, *pStatus* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BSigIn(ByVal *hPort* As Integer, ByVal *SigNo* As Short, ByRef *pStatus* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BSigIn(*hPort*: DWORD; *SigNo*: WORD; var *pStatus*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hPort　・・・ 汎用 I/O PORT ハンドルを指定します。

SigNo　・・・ 入力信号を指定します。

● C-770AL の汎用 I/O PORT を読み出す場合

引数 *SigNo* の値	指定信号
AC05_C770_IN0	$\overline{\text{IN0}}$
AC05_C770_IN1	$\overline{\text{IN1}}$

pStatus　・・・ 指定信号の状態が格納される変数のポインタを指定します。

格納される値	指定信号の状態
AC05_LOW	LOW レベル
AC05_HIGH	HIGH レベル

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT 一括書き込み関数

AL MCC05 ユニット対応 DLL

C-770AL

機　能

汎用 I/O PORT にデータを一括書き込みします。

書　式

C 言語　BOOL AC05_BPortOut(DWORD *hPort*, DWORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BPortOut(ByVal *hPort* As Long, *pData* As Long, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BPortOut(ByVal *hPort* As Integer, ByRef *pData* As Integer, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BPortOut(*hPort*: DWORD; var *pData*: DWORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hPort　・・・ 汎用 I/O PORT ハンドルを指定します。

pData　・・・ 書き込むデータが格納されている変数のポインタを指定します。
変数の内容は、次のようになります。

● C-770AL の汎用 I/O PORT に書き込む場合

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\overline{\text{OUT1}}$	$\overline{\text{OUT0}}$

HIGH レベルにするときに 0、LOW レベルにするときに 1 を書き込みます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT 指定信号 LOW レベル書き込み関数

AL MCC05 ユニット対応 DLL

C-770AL

機　能

汎用 I/O PORT の指定信号を LOW レベルにします。

書　式

C 言語　BOOL AC05_BSigLowOut(DWORD *hPort*, WORD *SigNo*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BSigLowOut(ByVal *hPort* As Long, ByVal *SigNo* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BSigLowOut(ByVal *hPort* As Integer, ByVal *SigNo* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BSigLowOut(*hPort*: DWORD; *SigNo*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hPort　・・・ 汎用 I/O PORT ハンドルを指定します。
SigNo　・・・ 出力信号を指定します。

● C-770AL の汎用 I/O PORT に書き込む場合

引数 *SigNo* の値	指定信号
AC05_C770_OUT0	$\overline{\text{OUT0}}$
AC05_C770_OUT1	$\overline{\text{OUT1}}$

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

汎用 I/O PORT 指定信号 HIGH レベル書き込み関数

AL MCC05 ユニット対応 DLL

C-770AL

機　能

汎用 I/O PORT の指定信号を HIGH レベルにします。

書　式

C 言語　BOOL AC05_BSigHighOut(DWORD *hPort*, WORD *SigNo*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BSigHighOut(ByVal *hPort* As Long, ByVal *SigNo* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BSigHighOut(ByVal *hPort* As Integer, ByVal *SigNo* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BSigHighOut(*hPort*: DWORD; *SigNo*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hPort　・・・ 汎用 I/O PORT ハンドルを指定します。
SigNo　・・・ 出力信号を指定します。

● C-770AL の汎用 I/O PORT に書き込む場合

引数 *SigNo* の値	指定信号
AC05_C770_OUT0	$\overline{OUT0}$
AC05_C770_OUT1	$\overline{OUT1}$

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

制御信号オープン関数

AL MCC05 **ユニット対応** DLL

CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

I/F 番号、スレーブアドレス、スレーブタイプを指定して、制御信号または制御 I/O PORT をオープンし、引数 *phSignal* で示される変数に制御信号ハンドルを格納します。

書　式

C 言語　BOOL AC05_BSignalOpen(WORD *IfNo*, WORD *SlaveAddr*, WORD *SlaveType*, DWORD FAR **phSignal*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BSignalOpen(ByVal *IfNo* As Integer, ByVal *SlaveAddr* As Integer, ByVal *SlaveType* As Integer, *phSignal* As Long, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BSignalOpen(ByVal *IfNo* As Short, ByVal *SlaveAddr* As Short, ByVal *SlaveType* As Short, ByRef *phSignal* As Integer, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BSignalOpen(*IfNo*: WORD; *SlaveAddr*: WORD; *SlaveType*: WORD; var *phSignal*: DWORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

IfNo　・・・ I/F 番号を指定します。

マスターのボード／ユニット	引数 *IfNo* の値
CB-09	ボード番号（0 ～ 9）
CB-23/USB	AC05_USB

SlaveAddr　・・・ スレーブアドレスを指定します。H'01 ～ H'1F のいずれかになります。

SlaveType　・・・ スレーブタイプを指定します。

引数 *SlaveType* の値	スレーブタイプ
AC05_SLAVE_CDB770	CDB-5420-AL770
AC05_SLAVE_CAD770	CAD-5410-AL770
AC05_SLAVE_C771	C-771
AC05_SLAVE_CD773	CD-773/ADB5331A

phSignal　・・・ 制御信号ハンドルが格納される変数のポインタを指定します。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

制御信号クローズ関数

AL MCC05 **ユニット対応** DLL

CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定された制御信号または制御 I/O PORT をクローズします。

書　式

C 言語　BOOL AC05_BSignalClose(DWORD *hSignal*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BSignalClose(ByVal *hSignal* As Long, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BSignalClose(ByVal *hSignal* As Integer, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BSignalClose(*hSignal*: DWORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hSignal　・・・ 制御信号ハンドルを指定します。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

制御信号読み出し関数

AL MCC05 ユニット対応 DLL

CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定された制御信号または制御 I/O PORT の状態を読み出します。

書　式

C 言語　BOOL AC05_BSignalIn(DWORD *hSignal*, DWORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BSignalIn(ByVal *hSignal* As Long, *pData* As Long, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BSignalIn(ByVal *hSignal* As Integer, ByRef *pData* As Integer, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BSignalIn(*hSignal*: DWORD; var *pData*: DWORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hSignal　・・・ 制御信号ハンドルを指定します。

pData　・・・ 読み出した内容が格納される変数のポインタを指定します。
変数の内容は、次のようになります。

● CDB-5420-AL770 の制御信号を読み出す場合

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	M.F

アクティブのときに 1、ノットアクティブのときに 0 が読み出されます。

● CAD-5410-AL770 の制御信号を読み出す場合

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	C.S	M.F

アクティブのときに 1、ノットアクティブのときに 0 が読み出されます。

● C-771 の制御 I/O PORT を読み出す場合

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	ADALM	ZDALM	YDALM	XDALM	$\overline{\text{AMF}}$ / $\overline{\text{AS.ON}}$	$\overline{\text{ZMF}}$ / $\overline{\text{ZS.ON}}$	$\overline{\text{YMF}}$ / $\overline{\text{YS.ON}}$	$\overline{\text{XMF}}$ / $\overline{\text{XS.ON}}$

アクティブのときに 1、ノットアクティブのときに 0 が読み出されます。

● CD-773/ADB5331A の制御 I/O PORT を読み出す場合

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\overline{\text{YMF}}$	$\overline{\text{XMF}}$

アクティブのときに 1、ノットアクティブのときに 0 が読み出されます。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

制御信号指定ビット読み出し関数

AL MCC05 ユニット対応 DLL

CAD-5410-AL770 | C-771

機　能

指定された制御信号または制御 I/O PORT の指定ビットの状態を読み出します。

書　式

C 言語　BOOL AC05_BSignalBitIn(DWORD *hSignal*, WORD *SigNo*, WORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BSignalBitIn(ByVal *hSignal* As Long, ByVal *SigNo* As Integer, *pData* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BSignalBitIn(ByVal *hSignal* As Integer, ByVal *SigNo* As Short, ByRef *pData* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BSignalBitIn(*hSignal*: DWORD; *SigNo*: WORD; var *pData*: WORD; var *psResult*: AC05_S_RESULT) : Boolean;

引　数

hSignal　・・・ 制御信号ハンドルを指定します。
SigNo　・・・ 制御信号のビットを指定します。

● CAD-5410-AL770 の制御信号を読み出す場合

引数 *SigNo* の値	ビット
AC05_CAD770_MF	M.F
AC05_CAD770_CS	C.S

● C-771 の制御 I/O PORT を読み出す場合

引数 *SigNo* の値	ビット
AC05_C771_XDALM	XDALM
AC05_C771_YDALM	YDALM
AC05_C771_ZDALM	ZDALM
AC05_C771_ADALM	ADALM

pData　・・・ 読み出した内容が格納される変数のポインタを指定します。

格納される値	ビットの状態
AC05_ON	ON
AC05_OFF	OFF

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

制御信号書き込み関数

AL MCC05 ユニット対応 DLL

CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

指定された制御信号または制御 I/O PORT にデータを書き込みます。

書　式

C 言語　BOOL AC05_BSignalOut(DWORD *hSignal*, DWORD FAR **pData*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BSignalOut(ByVal *hSignal* As Long, *pData* As Long, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BSignalOut(ByVal *hSignal* As Integer, ByRef *pData* As Integer, ByRef *psResult* As AC05_S_RESULT)As Boolean

Delphi　function AC05_BSignalOut(*hSignal*: DWORD; var *pData*: DWORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hSignal　・・・ 制御信号ハンドルを指定します。

pData　・・・ 書き込むデータが格納される変数のポインタを指定します。
変数の内容は、次のようになります。

● CDB-5420-AL770 の制御信号にデータを書き込む場合

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	M.F

1 を書き込むとアクティブ、0 を書き込むとノットアクティブになります。

● CAD-5410-AL770 の制御信号にデータを書き込む場合

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	C.S	M.F

1 を書き込むとアクティブ、0 を書き込むとノットアクティブになります。

● C-771 の制御 I/O PORT にデータを書き込む場合

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	0	0	$\overline{\text{AMF}}$ / $\overline{\text{AS.ON}}$	$\overline{\text{ZMF}}$ / $\overline{\text{ZS.ON}}$	$\overline{\text{YMF}}$ / $\overline{\text{YS.ON}}$	$\overline{\text{XMF}}$ / $\overline{\text{XS.ON}}$

1 を書き込むとアクティブ、0 を書き込むとノットアクティブになります。

● CD-773/ADB5331A の制御 I/O PORT にデータを書き込む場合

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\overline{\text{YMF}}$	$\overline{\text{XMF}}$

1 を書き込むとアクティブ、0 を書き込むとノットアクティブになります。

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

制御信号指定ビット ON 書き込み関数

AL MCC05 ユニット対応 DLL

CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

制御信号または制御 I/O PORT の指定されたビットを ON にします。

書　式

C 言語　BOOL AC05_BSignalBitOnOut(DWORD *hSignal*, WORD *SigNo*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BSignalBitOnOut(ByVal *hSignal* As Long, ByVal *SigNo* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BSignalBitOnOut(ByVal *hSignal* As Integer, ByVal *SigNo* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BSignalBitOnOut(*hSignal*: DWORD; *SigNo*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hSignal　・・・ 制御信号ハンドルを指定します。
SigNo　・・・ 制御信号のビットを指定します。

● CAD-5410-AL770 の制御信号の指定ビットを ON にする場合

引数 *SigNo* の値	ビット
AC05_CAD770_MF	M.F
AC05_CAD770_CS	C.S

● C-771 の制御 I/O PORT の指定ビットを ON にする場合

引数 *SigNo* の値	ビット
AC05_C771_XMF_SON	$\overline{\text{XMF}}$ または $\overline{\text{XS.ON}}$
AC05_C771_YMF_SON	$\overline{\text{YMF}}$ または $\overline{\text{YS.ON}}$
AC05_C771_ZMF_SON	$\overline{\text{ZMF}}$ または $\overline{\text{ZS.ON}}$
AC05_C771_AMF_SON	$\overline{\text{AMF}}$ または $\overline{\text{AS.ON}}$

● CD-773/ADB5331A の制御 I/O PORT の指定ビットを ON にする場合

引数 *SigNo* の値	ビット
AC05_CD773_XMF	$\overline{\text{XMF}}$
AC05_CD773_YMF	$\overline{\text{YMF}}$

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

制御信号指定ビット OFF 書き込み関数

AL MCC05 ユニット対応 DLL

CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

制御信号または制御 I/O PORT の指定されたビットを OFF にします。

書　式

C 言語　BOOL AC05_BSignalBitOffOut(DWORD *hSignal*, WORD *SigNo*, AC05_S_RESULT FAR **psResult*);

VB　Function AC05_BSignalBitOffOut(ByVal *hSignal* As Long, ByVal *SigNo* As Integer, *psResult* As AC05_S_RESULT) As Boolean

VB.NET　Function AC05_BSignalBitOffOut(ByVal *hSignal* As Integer, ByVal *SigNo* As Short, ByRef *psResult* As AC05_S_RESULT) As Boolean

Delphi　function AC05_BSignalBitOffOut(*hSignal*: DWORD; *SigNo*: WORD; var *psResult*: AC05_S_RESULT): Boolean;

引　数

hSignal　・・・ 制御信号ハンドルを指定します。
SigNo　・・・ 制御信号のビットを指定します。

● CAD-5410-AL770 の制御信号の指定ビットを OFF にする場合

引数 *SigNo* の値	ビット
AC05_CAD770_MF	M.F
AC05_CAD770_CS	C.S

● C-771 の制御 I/O PORT の指定ビットを OFF にする場合

引数 *SigNo* の値	ビット
AC05_C771_XMF_SON	$\overline{\text{XMF}}$ または $\overline{\text{XS.ON}}$
AC05_C771_YMF_SON	$\overline{\text{YMF}}$ または $\overline{\text{YS.ON}}$
AC05_C771_ZMF_SON	$\overline{\text{ZMF}}$ または $\overline{\text{ZS.ON}}$
AC05_C771_AMF_SON	$\overline{\text{AMF}}$ または $\overline{\text{AS.ON}}$

● CD-773/ADB5331A の制御 I/O PORT の指定ビットを OFF にする場合

引数 *SigNo* の値	ビット
AC05_CD773_XMF	$\overline{\text{XMF}}$
AC05_CD773_YMF	$\overline{\text{YMF}}$

psResult　・・・ この関数を実行した結果が格納される RESULT 構造体のポインタを指定します。
NULL ポインタまたは 0 が指定されると、実行結果が格納されません。

戻り値

この関数を実行した結果、正常終了したときは TRUE(1)、エラーが発生したときは FALSE(0)を返します。

データセット関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

24 ビットデータを、次の形式で、データ構造体に格納します。

引数 *psData* で示されるデータ構造体のメンバ *AC05_Data[0]*（C 言語表記）［各種 DATA1 PORT に対応］

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	2^{23}	←	引数 *Data* の 2^{23} ～ 2^{16}				→	2^{16}

引数 *psData* で示されるデータ構造体のメンバ *AC05_Data[1]*（C 言語表記）［各種 DATA2 PORT に対応］

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	2^{15}	←	引数 *Data* の 2^{15} ～ 2^{8}				→	2^{8}

引数 *psData* で示されるデータ構造体のメンバ *AC05_Data[2]*（C 言語表記）［各種 DATA3 PORT に対応］

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
0	0	0	0	0	0	0	0	2^{7}	←	引数 *Data* の 2^{7} ～ 2^{0}				→	2^{0}

書　式

C 言語　VOID AC05_SetData(DWORD *Data*, AC05_S_DATA FAR **psData*);

VB　Sub AC05_SetData(ByVal *Data* As Long, *psData* As AC05_S_DATA)

VB.NET　Sub AC05_SetData(ByVal *Data* As Integer, ByRef *psData* As AC05_S_DATA)

Delphi　procedure AC05_SetData(*Data*: DWORD; var *psData*: AC05_S_DATA);

引　数

Data　・・・ 24 ビットのデータを指定します。下位 24 ビットが有効です。
psData　・・・ データ構造体のポインタを指定します。

戻り値

この関数に、戻り値はありません。

データゲット関数

AL MCC05 ユニット対応 DLL

C-770AL | CDB-5420-AL770 | CAD-5410-AL770 | C-771 | CD-773/ADB5331A

機　能

データ構造体の内容を、次の形式で、24 ビットデータに変換します。

変換後の 24 ビットデータ

2^{31}	2^{30}	2^{29}	2^{28}	2^{27}	2^{26}	2^{25}	2^{24}	2^{23}	2^{22}	2^{21}	2^{20}	2^{19}	2^{18}	2^{17}	2^{16}
下位 24 ビットを符号拡張								2^7 ←	メンバ *AC05_Data[0]* ［各種 DATA1 PORT に対応］					→	2^0

2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
2^7 ←	メンバ *AC05_Data[1]* ［各種 DATA2 PORT に対応］					→	2^0	2^7 ←	メンバ *AC05_Data[2]* ［各種 DATA3 PORT に対応］					→	2^0

メンバは C 言語表記です

書　式

C 言語　DWORD AC05_GetData(AC05_S_DATA FAR **psData*);

VB　Function AC05_GetData(*psData* As AC05_S_DATA) As Long

VB.NET　Function AC05_GetData(ByRef *psData* As AC05_S_DATA) As Integer

Delphi　function AC05_GetData(var *psData*: AC05_S_DATA): DWORD;

引　数

psData　・・・ データ構造体のポインタを指定します。

戻り値

24 ビットに変換されたデータを返します。

6. ソフト開発に必要なファイル

ユーザアプリケーション開発に必要なファイルは、インストール時に指定する次のフォルダに格納されています。
(インストール時にパスを\ Program Files 指定した場合)

● AL I/F ユニット対応 DLL を用いたソフト開発に必要なファイル

ファイル	ファイルの場所
Visual C++ヘッダファイル	\ Program Files\ Mpl28\ Bin\ Vc\ AlUsbA.h
Visual C++ライブラリファイル	\ Program Files\ Mpl28\ Bin\ Vc\ VcAlUsbA.lib
C++ Builder ヘッダファイル	\ Program Files\ Mpl28\ Bin\ Builder\ AlUsbA.h
C++ Builder ライブラリファイル	\ Program Files\ Mpl28\ Bin\ Builder\ BcAlUsbA.lib
Visual Basic .NET 関数定義ファイル	\ Program Files\ Mpl28\ Bin\ Vb.NET\ AlUsbA.vb
Visual Basic 関数定義ファイル	\ Program Files\ Mpl28\ Bin\ Vb\ AlUsbA.bas

● AL I/O ユニット対応 DLL を用いたソフト開発に必要なファイル
AL I/O ユニット取扱説明書をご覧ください

● AL MCC05 ユニット対応 DLL を用いたソフト開発に必要なファイル

ファイル	ファイルの場所
Visual C++ヘッダファイル	\ Program Files\ Mpl28\ Bin\ Vc\ Ac05A.h
Visual C++ライブラリファイル	\ Program Files\ Mpl28\ Bin\ Vc\ VcAc05A.lib
C++ Builder ヘッダファイル	\ Program Files\ Mpl28\ Bin\ Builder\ Ac05A.h
C++ Builder ライブラリファイル	\ Program Files\ Mpl28\ Bin\ Builder\ BcAc05A.lib
Visual Basic .NET 関数定義ファイル	\ Program Files\ Mpl28\ Bin\ Vb.NET\ Ac05A.vb
Visual Basic 関数定義ファイル	\ Program Files\ Mpl28\ Bin\ Vb\ Ac05A.bas

● AL MCC06 ユニット対応 DLL を用いたソフト開発に必要なファイル
AL MCC06 ユニット取扱説明書をご覧ください

7. サンプルプログラム

7-1. 概要

Visual C++ .NET、Visual C++、C++ Builder、Visual Basic .NET、Visual Basic のサンプルプログラムが用意されています。
サンプルプログラムのファイルは、インストール時に指定する次のフォルダに格納されています。
（インストール時にパスを Program Files と指定した場合）

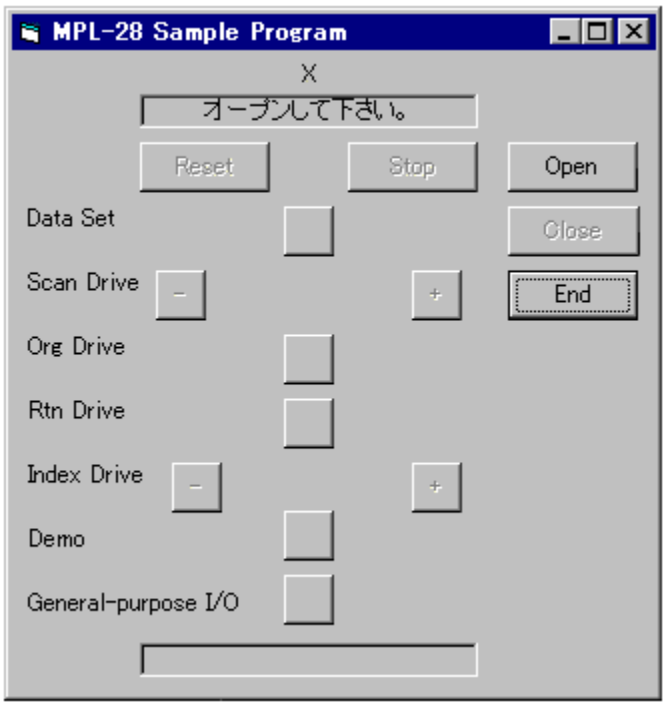
言語	ファイルの場所
Visual C++ .NET、Visual C++	\ Program Files\ Mpl28\ Sample\ Vc
C++ Builder	\ Program Files\ Mpl28\ Sample\ Builder
Visual Basic .NET	\ Program Files\ Mpl28\ Sample\ Vb.NET
Visual Basic	\ Program Files\ Mpl28\ Sample\ Vb

7-2. ご使用になる前に

環境設定ツールで AL 通信ボーレート、リトライ回数を設定して下さい。
パソコン、CB-23/USB、C-772（SLAVE ADDRESS = H'01）、CB-08（SLAVE ADDRESS = H'02）を接続して下さい。

7-3. 仕様

サンプルプログラムには、Visual C++ 6.0、C++ Builder 5.0、Visual Basic .NET 2002、Visual Basic 6.0 で作成したものを用意してあります。これらは、同じ仕様で作られています。
サンプルプログラムを参照する場合には、それぞれの言語の開発環境からプロジェクトを開いて下さい。



Open ボタン	デバイス(C-772 X 軸)をオープンします。
Close ボタン	デバイス(C-772 X 軸)をクローズします。
End ボタン	サンプルプログラムを終了します。
Reset ボタン	ADDRESS COUNTER を 0 に PRESET します。
Stop ボタン	DRIVE を即時停止します。
Data Set ボタン	次の設定にします。 RATE TYPE : L1-TYPE URATE : 10ms/1000Hz LSPD : 1000Hz DRATE : 10ms/1000Hz HSPD : 5000Hz
Scan Drive +ボタン	+(CW)方向へ SCAN DRIVE します。
Scan Drive -ボタン	-(CCW)方向へ SCAN DRIVE します。
Org Drive ボタン	機械原点検出形式 ORG-3 で ORIGIN DRIVE を行います。
Rtn Drive ボタン	絶対 ADDRESS 0 へ移動する ABSOLUTE INDEX DRIVE を行います。
Index Drive +ボタン	+(CW)方向へ 3000 パルス移動する INCREMENTAL INDEX DRIVE を行います。
Index Drive -ボタン	-(CCW)方向へ 3000 パルス移動する INCREMENTAL INDEX DRIVE を行います。
Demo ボタン	次の動作を連続して行います。 機械原点の検出(ORIGIN DRIVE) 電気原点の設定(ADDRESS COUNTER を 0 に PRESET) +(CW)方向へ 4000 パルス移動を 4 回繰り返す(INCREMENTAL INDEX DRIVE) 絶対 ADDRESS 30000 へ移動(ABSOLUTE INDEX DRIVE) 電気原点へ移動(ABSOLUTE INDEX DRIVE)
General-Purpose I/O ボタン	次の動作を連続して行います。 I/O PORT OPEN(CB-08) IN10 - IN12 が HIGH になるまで待つ OUT10 - OUT12 を HIGH にする +(CW)方向へ 1000 パルス移動(INCREMENTAL INDEX DRIVE) OUT10 を LOW にする IN11 が LOW になるまで待つ +(CW)方向へ 2000 パルス移動(INCREMENTAL INDEX DRIVE) OUT11 を LOW にする IN12 が LOW になるまで待つ +(CW)方向へ 3000 パルス移動(INCREMENTAL INDEX DRIVE) OUT12 を LOW にする I/O PORT CLOSE(CB-08)

8. トラブルシューティング

作成したアプリケーションプログラムが正常に動作しない場合、次のことを行って下さい。

(1) I/F ユニットの認識

スレーブ情報確認プログラムを起動して下さい。
スレーブ情報確認プログラムは、起動時に I/F ユニットの認識を確認しています。

(2) スレーブの認識

スレーブ情報確認プログラムは、実際に USB 通信、AL 通信を行い、各スレーブアドレスに接続されている
スレーブを検出し、検出結果を表示しています。

メッセージ	スレーブの検出結果
Unconnection	スレーブが接続されていないことを示しています。
C-770AL	C-770AL が接続されていることを示しています。
CB-08	CB-08 が接続されていることを示しています。
CDB-5420-AL770	CDB-5420-AL770 が接続されていることを示しています。
CAD-5410-AL770	CAD-5410-AL770 が接続されていることを示しています。
C-771	C-771 が接続されていることを示しています。
CD-773/ADB5331A	CD-773/ADB5331A が接続されていることを示しています。
C-772	C-772 が接続されていることを示しています。
CB-34	CB-34 が接続されていることを示しています。

(3) RESULT 構造体の確認

各関数は、アプリケーションプログラムによって与えられた引数の内容をチェックし、エラーがある場合は、
FALSE(0)を返し、正常である場合は TRUE(1)を返します。
関数が正常に動作していないと思われるステップの後にブレークポイントを設定し、関数が返した値が TRUE(1)
であることを確認して下さい。
TRUE(1)でない場合は、エラー原因を特定する為に RESULT 構造体の内容を参照して下さい。

本版で改訂された主な箇所

箇　所	内　容
なし	

■ 製品保証

保証期間と保証範囲について

● 納入品の保証期間は、納入後１ヶ年と致します。
● 上記保証期間中に当社の責により故障を生じた場合は、その修理を当社の責任において行います。
(日本国内のみ)
ただし、次に該当する場合は、この保証対象範囲から除外させて頂きます。
(1) お客様の不適当な取り扱い、ならびに使用による場合。
(2) 故障の原因が、当製品以外からの事由による場合。
(3) お客さまの改造、修理による場合。
(4) 製品出荷当時の科学・技術水準では予見が不可能だった事由による場合。
(5) その他、天災、災害等、当社の責にない場合。
(注1)ここでいう保証は、納入品単体の保証を意味するもので、納入品の故障により誘発される損害はご容赦頂きます。
(注2)当社において修理済みの製品に関しましては、保証外とさせて頂きます。

技術相談のお問い合わせ

TEL.(042)664-5382 FAX.(042)666-5664
E-mail s-support@melec-inc.com

販売に関するお問い合わせ

TEL.(042)664-5384 FAX.(042)666-2031

株式会社メレック 制御機器営業部
〒193-0834 東京都八王子市東浅川町516-10 **URL:http://www.melec-inc.com**

記載内容は、製品改良のため予告なく変更することがありますのでご了承ください。